Report

I made the insertAfter function in the main file, declared it before the main function, and defined it after the main function. In insertAfter I sent three parameters, both of the values and the vector, I used the iterator to loop (traverse) the vector and check for the first value. After finding the value, I inserted the second value right after it. For the linked list class implementation, I did a struct for the node and a class for the linked list the class had a pointer of type struct (node) to point at the first node, and each struct had a pointer pointing at the next node and used my notes from the lecture as a reference. For the add function, I allowed it to add to the end of the list. For the remove function, I made it remove a specific node by sending the value of the node I want to remove to the function. In the print function, I made a while loop to traverse the list and print the nodes. In the create list function, I sent a vector to the function and used nested for loops to check if there were multiple occurrences of a value inside of the vector and counted the occurrences using a count variable that reset at the beginning of each new value, I used “this” operator to add a new node for each value of the vector. finally, for the sum function, I traversed the list using a while loop and calculated and returned the sum. In the main, I ask the user about the vector size and then ask the user to input the integers of the vector. And then asked the user to insert the first value that they wanted to add another value after and allowed them to input the first and second value, respectively. I made an instance of the linked list class and called for create list function, and sent the vector in it. Finally, I called the print elements function.

screenshot of the output:

```
How many integers do you want to add ?
5
Please enter the 5 numbers:
1
2
3
4
5
to insert a value after another
please enter the first value and second value respectively:
5
4


linked list of vector elements
1 occured 1 time
2 occured 1 time
3 occured 1 time
4 occured 2 time
5 occured 1 time
4 occured 2 time
Program ended with exit code: 0
```